Quantum algorithm for the Boolean hidden shift problem

Dmitry Gavinsky*, Martin Roetteler*, and Jérémie Roland*

NEC Laboratories America, Inc.

Abstract. The hidden shift problem is a natural place to look for new separations between classical and quantum models of computation. One advantage of this problem is its flexibility, since it can be defined for a whole range of functions and a whole range of underlying groups. In a way, this distinguishes it from the hidden subgroup problem where more stringent requirements about the existence of a periodic subgroup have to be made. And yet, the hidden shift problem proves to be rich enough to capture interesting features of problems of algebraic, geometric, and combinatorial flavor. We present a quantum algorithm to identify the hidden shift for any Boolean function. Using Fourier analysis for Boolean functions we relate the time and query complexity of the algorithm to an intrinsic property of the function, namely its minimum influence. We show that for randomly chosen functions the time complexity of the algorithm is polynomial. Based on this we show an average case exponential separation between classical and quantum time complexity. A perhaps interesting aspect of this work is that, while the extremal case of the Boolean hidden shift problem over so-called bent functions can be reduced to a hidden *subgroup* problem over an abelian group, the more general case studied here does not seem to allow such a reduction.

1 Introduction

Hidden shift problems have been studied in quantum computing as they provide a framework that can give rise to new quantum algorithms. The hidden shift problem was first introduced and studied in a paper by van Dam, Hallgren and Ip [vDHI06] and is defined as follows. We are given two functions f, g that map a finite group G to some set with the additional promise that there exists an element $s \in G$, the so-called shift, such that for all x it holds that $g(x) = f(x+s)$. The task is to find s. Here the group G is additively denoted, but the problem can be defined for non-abelian groups as well. The great flexibility in the definition allows to capture interesting problems ranging from algebraic problems such as the shifted Legendre symbol [vDHI06], over geometric problems such as finding the center of shifted spheres [CSV07, Liu09] and shifted lattices [Reg04], to combinatorial problems such as graph isomorphism [CW07].

* {dmitry,mroetteler,jroland}@nec-labs.com

Notable here is a well-known connection between the hidden subgroup problem for the dihedral group, a notoriously difficult instance which itself has connections to lattice problems and average case subset sum [Reg04] and a hidden shift problem over the cyclic group $\mathbb{Z}_n$ where the functions f and g are injective [Kup05, MRRS07, CvD07]. It is known [FIM+03, Kup05] that the hidden shift problem for *injective* functions $f, g : G \to S$ that map from an abelian G to a set S is equivalent to hidden subgroup problem over the semi-direct product between G and $\mathbb{Z}_2$, where the action of $\mathbb{Z}_2$ on G is given by the inverse. We would like to point out that the functions studied here are Boolean functions (i.e., $G = \mathbb{Z}_2^n$) and therefore far from being injective. Even turning them into injective *quantum* functions, as is possible for bent functions [Röt10], seems not to be obvious in this case. Another recent example of a non-abelian hidden shift problem arises in a reduction used to argue that the McEliece cryptosystems withstands certain types of quantum attacks [DMR10].

In this paper we confine ourselves to the abelian case and in particular to the case where $G = \mathbb{Z}_2^n$ is the Boolean hypercube. The resulting hidden shift problem for Boolean functions, i.e., functions that take n bits as inputs and output just 1 bit, at first glance looks rather innocent. However, to our knowledge, the Boolean case was previously only addressed for two extreme cases: a) functions which mark precisely one element and b) functions which are maximally apart from any affine Boolean function (so-called bent functions). In case a), the problem of finding the shift is the same as unstructured search, so that the hidden shift can be found by Grover's algorithm [Gro96] and the query complexity is known to be tight and is given by $\Theta(\sqrt{2^n})$.

In case b) the hidden shift can be discovered in one query using an algorithm that was found by one of the co-authors [Röt10], provided that the *dual* of the function can be computed efficiently, where the definition of the dual is via the Fourier spectrum of the function which in this case can be shown to be flat in absolute value. If no efficient implementation of the dual is known then still a quantum algorithm exists that can identify the hidden shift in $O(n)$ queries. The present paper can be thought of as a generalization of this latter algorithm to the case of Boolean functions other than those having a flat spectrum. This is motivated by the quite natural question of what happens when the extremal conditions leading to the family of bent functions are relaxed. In this paper we address the question of whether there is a broader class of functions for which hidden shifts of a function can be identified.

The first obvious step in direction of a generalization is actually a roadblock: Grover's search problem [Gro96] can also be cast as a hidden shift problem. In this case the corresponding class of Boolean functions are the *delta functions*, i.e., $f, g : \{0,1\}^n \to \{0,1\}$, where $g(x) = f(x+s)$ and $f(x)$ is the function that takes value 1 on input $(0, \ldots, 0)$ and 0 elsewhere and $g(x)$ is the function that takes the value 1 on input s and 0 elsewhere. Grover's algorithm [Gro96] allows to find s in time $O(\sqrt{2^n})$ on a quantum computer (which is also the fastest possible [BV97]).

Thus, the following situation emerges for the quantum and the classical query complexities of these two extremal cases: for bent functions the classical query complexity[1] is $\Omega(\sqrt{2^n})$ and the quantum query complexity[2] is $O(n)$. For delta functions the classical query complexity is $\Theta(2^n)$ and the quantum query complexity is $\Theta(\sqrt{2^n})$.

For a general Boolean function the hidden shift problem can be seen as lying somewhere between these two extreme cases. This is somewhat similar to how the so-called weighing matrix problem [vD08] interpolates between the Bernstein-Vazirani problem [BV97] and Grover search, and how the generalized hidden shift problem [CvD07] interpolates between the abelian and dihedral hidden subgroup problems. However, apart from these two extremes, not much is known about the query complexity of the hidden shift problem for general Boolean functions.

The main goal of this work was to understand the space between these two extremes. We show that there is a natural way to "interpolate" between them and to give an algorithm for each Boolean function whose query complexity depends only on properties of the Fourier spectrum of that function.

Prior work. As far as hidden shifts of Boolean functions are concerned, besides the mentioned papers about the bent case and the case of search, very little was known. The main technique previously used to tackle hidden shift problem was by computing a suitable convolution. However, in order to maintain unitarity, much of target function's features that we want to compute the convolution with had to be "sacrificed" by requiring the function to become diagonal unitary, leading to a renormalization of the diagonal elements, an issue perhaps first pointed out by [CM04]. No such renormalization is necessary if the spectrum is already flat which corresponds to the case of the Legendre symbol [vDHI06] (with the exception of one special value at 0) and the case of bent functions which was considered in [Röt10].

Our results. We introduce a quantum algorithm that allows us to sample from vectors that are perpendicular to the hidden shift v according to a distribution that is related to the Fourier spectrum of the given Boolean function f. If f is bent, then this distribution is uniform which in turn leads to a unique characterization of v from $O(n)$ queries via a system of linear equations. For general f more queries might be necessary and intuitively the more concentrated the Fourier spectrum of f is, the more queries have to be made: in the extreme case of a (± 1 valued) delta function f the spectrum is extremely imbalanced and concentrated almost entirely on the zero Fourier coefficient which corresponds

[1] Note that the query complexity depends crucially on how the functions f and g can be accessed: the stated bounds hold for the case where f and g are given as black-boxes. If f is a *known* bent function, then it is easy to see that the classical query complexity becomes $O(n)$.

[2] A further improvement is possible in case the so-called *dual* bent function $\widetilde{f}$ is accessible via another black-box: in this case the quantum query complexity becomes constant [Röt10].

to the case of unstructured search for which our algorithm offers no advantage over Grover's algorithm. For general f we give an upper bound on the number of queries in terms of the *influence* γ_f of the function f, where the influence is defined as $\gamma_f = \min_v(\mathbf{Pr}_x[f(x) \neq f(x+v)])$.

From a simple application of the Chernoff bound it follows that it is extremely unlikely that a randomly chosen Boolean function will give rise to a hard instance for our quantum algorithm. This in turn gives rise to our main result of the paper:

Theorem 2 (Average case exponential separation). *Let $(\mathcal{O}_f, \mathcal{O}_g)$ be an instance of a Boolean hidden shift problem (BHSP) where $g(x) = f(x+v)$ and f and v are chosen uniformly at random. Then there exists a quantum algorithm which finds v with bounded error using $O(n)$ queries and in $O(\mathrm{poly}(n))$ time whereas any classical algorithm needs $\Omega(2^{n/2})$ queries to achieve the same task.*

This result can be interpreted as an exponential quantum-classical separation for the time and query complexity of an average case problem. Finally, we would like to comment on the relationship between the problem considered in this paper and the abelian hidden subgroup problem. It is interesting to note, yet not particularly difficult to see, that the case of a hidden shift problem for bent functions can be reduced to that of an abelian hidden subgroup problem. The hiding function in this case is a quantum function, i. e., it takes values in the set of quantum sets rather than just basis states. For the case of a non-bent function, including the cases of random functions considered here, the same direct correspondence to the hidden subgroup problem over an abelian group no longer exists, i. e., even though there is no obvious group/subgroup structure present in the function f, the algorithm can still identify the hidden shift v.

2 Preliminaries

Definition 1 (Boolean Hidden Shift Problem). *Let $n \geq 1$ and let $f, g : \mathbb{Z}_2^n \to \mathbb{Z}_2$ be two Boolean functions such that the following conditions hold:*

- *if for some $t \in \mathbb{Z}_2^n$ it holds that $f(x) \equiv f(x+t)$ then $t = 0$;*
- *for some $s \in \mathbb{Z}_2^n$ it holds that $g(x) \equiv f(x+s)$.*

If f and g are given by two oracles O_f and O_g, we say that the pair (O_f, O_g) defines an instance of a hidden shift problem (BHSP) for the function f. The value $s \in \mathbb{Z}_2^n$ that satisfies $g(x) \equiv f(x+s)$ is the solution of the given instance of the BHSP.

We also consider the $\{+1, -1\}$-valued function F corresponding to the function f and view it as a function over $\mathbb{R}$, that is,

$$F : \mathbb{Z}_2^n \to \mathbb{R} : x \mapsto (-1)^{f(x)}. \tag{1}$$

The arguments of these functions are assumed to belong to $\mathbb{Z}_2^n$, and their *inner product* is defined accordingly, i.e., $\langle u, v \rangle = \bigoplus_{i=1}^n u_i \cdot v_i$. We also denote by

$\chi_u(\dots)$ the elements of the standard Fourier basis corresponding to $\mathbb{Z}_2^n$, that is, $\chi_u(v) = (-1)^{\langle u,v\rangle}$ for every $u, v \in \mathbb{Z}_2^n$.

We will see that the complexity of the BHSP depends on the notion of influence.

Definition 2 (Influence). *For any Boolean function f over $\mathbb{Z}_2^n$ and n-bit string v, we call $\gamma_{f,v} = \mathbf{Pr}_x[f(x) \neq f(x+v)]$ the influence of v over f, and $\gamma_f = \min_v \gamma_{f,v}$ the minimum influence over f.*

The following lemma relates the influence over a Boolean function f to the Fourier spectrum of its $\{+1,-1\}$-valued analog F, see also [GOS+09, Fact 11, p. 14].

Lemma 1. $\gamma_{f,v} = \sum_{u:\langle v,u\rangle=1} \left|\widehat{F}(u)\right|^2$.

We give a proof of this lemma in Appendix A for completeness.

3 Our algorithm

Theorem 1. *There exists a quantum algorithm that solves an instance of BHSP defined over the function f using expected $O(n/\sqrt{\gamma_f})$ oracle queries. The algorithm takes expected time polynomial in the number of queries.*

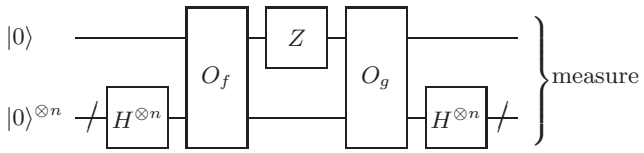

Fig. 1. Quantum circuit for the **Sampling Subroutine**.

Proof. The algorithm relies on the **Sampling Subroutine** described in Fig. 1, where H denotes the standard Hadamard gate, Z is a phase gate acting on one qubit as $Z : |b\rangle \mapsto (-1)^b|b\rangle$, and O_f is the oracle for f acting on $n+1$ qubits as $O_f : |b\rangle|x\rangle \mapsto |b \oplus f(x)\rangle|x\rangle$ (similarly for O_g). The algorithm works as follows:

Quantum algorithm

1. Set $i = 1$
2. Run the **Sampling Subroutine**. Denote by (b_i, u_i) the output of the measurement.
3. If $\mathsf{Span}\{u_k | k \in [i]\} \neq \mathbb{Z}_2^n$, increment $i \to i+1$ and go back to step 2. Otherwise set $t = i$ and continue.
4. Output "s", where s is the unique solution of

$$\begin{cases} \langle u_1, s\rangle = b_1; \\ \quad \dots \\ \langle u_t, s\rangle = b_t. \end{cases}$$

Obviously, this algorithm makes $O(t)$ quantum queries to the oracles and its complexity is polynomial in $t+n$. The quantum state before the measurement is

$$\begin{aligned}
|0\rangle|0\rangle^{\otimes n} &\overset{H^{\otimes n}}{\longmapsto} \frac{1}{\sqrt{2^n}}\sum_x |0\rangle|x\rangle \overset{O_f}{\longmapsto} \frac{1}{\sqrt{2^n}}\sum_x |f(x)\rangle|x\rangle \overset{Z}{\longmapsto} \frac{1}{\sqrt{2^n}}\sum_x (-1)^{f(x)}|f(x)\rangle|x\rangle \\
&\overset{O_g}{\longmapsto} \frac{1}{\sqrt{2^n}}\sum_x (-1)^{f(x)}|f(x)\oplus g(x)\rangle|x\rangle \\
&= \frac{1}{\sqrt{2^n}}|0\rangle\sum_x \frac{F(x)+F(x+s)}{2}|x\rangle + \frac{1}{\sqrt{2^n}}|1\rangle\sum_x \frac{F(x)-F(x+s)}{2}|x\rangle \\
&\overset{H^{\otimes n}}{\longmapsto} |0\rangle\sum_u \frac{1+\chi_u(s)}{2}\widehat{F}(u)|u\rangle + |1\rangle\sum_u \frac{1-\chi_u(s)}{2}\widehat{F}(u)|u\rangle. \qquad (2)
\end{aligned}$$

Its measurement therefore always returns a pair $(b_i, u_i) \in \{0,1\}\times\{0,1\}^n$ where $\langle u_i, s\rangle = b_i$. Moreover, since by construction $\mathsf{Span}\{u_i | i\in[t]\} = \mathbb{Z}_2^n$, the system of equations in step 4 accepts a unique solution that can only be the hidden shift s, thus the final answer of our algorithm is always correct.

We now show that the algorithm terminates in bounded expected time. We need to prove that repeatedly sampling using the procedure in step 2 yields n linearly independent vectors u_i, therefore spanning $\mathbb{Z}_2^n$, after a bounded expected number of trials t. Let (B,U) be a pair of random variables describing the measurement outcomes for the **Sampling Subroutine**, and D_f^U denote the marginal distribution of U. From the right-hand side of (2) it is clear that

$$D_f^U(u) \equiv \left|\widehat{F}(u)\right|^2.$$

Note that this distribution does not depend on g.

Let d_i be the dimension of $\mathsf{Span}\{u_k | k\in[i]\}$. By construction, we have $d_1 = 1, d_t = n$ and d_{i+1} equals either d_i or d_i+1. Let us bound the probability that $d_{i+1} = d_i+1$, or, equivalently, that $u_{i+1} \notin \mathsf{Span}\{u_k | k\in[i]\}$. This probability can only decrease as d_i increases, so let us consider the worst case where $d_i = n-1$. In that case, there exists some $v \in \mathbb{Z}_2^n \setminus \{0\}$ such that $\mathsf{Span}\{u_k | k\in[i]\}$ is exactly the subspace orthogonal to v. Then, the probability that u_{i+1} distributed according to D_f^U does not lie in this subspace (and hence $d_{i+1} = d_i+1$) is given by

$$\mathop{\mathbf{Pr}}_{u\sim D_f^U}[\langle v,u\rangle = 1] = \sum_{u:\langle v,u\rangle=1}\left|\widehat{F}(u)\right|^2 = \gamma_{f,v},$$

which follows from Lemma 1. Therefore, for any i, the probability that $d_{i+1} = d_i+1$ is at least $\gamma_f = \min_v \gamma_{f,v}$, and the expected number of trials before it happens is at most $1/\gamma_f$. Since d_i must be incremented n times, the expected total number of trials t is at most n/γ_f.

Using quantum amplitude amplification, we can obtain a quadratic improvement over this expected running time. Indeed, instead of repeating the **Sampling Subroutine** $O(1/\gamma_f)$ times until we obtain a sample u not in the subspace

spanned by the previous samples, we can use quantum amplitude amplification, which achieves the same goal using only $O(1/\sqrt{\gamma_f})$ applications of the quantum circuit in the **Sampling Subroutine** (see [BHMT02, Theorem 3]). We therefore obtain a quantum algorithm that solves the problem with success probability 1 and an expected number of queries $O(n/\sqrt{\gamma_f})$. □

In case a lower bound on γ_f is known, we have the following corollary:

Corollary 1. *There exists a quantum algorithm that solves an instance of BHSP defined over the function f, with the promise that $\gamma_f \geq \delta$, with success probability at least $1-\varepsilon$ and using at most $O(n\log(1/\varepsilon)/\sqrt{\delta})$ oracle queries. The algorithm takes expected time polynomial in the number of queries.*

Proof. This immediately follows from Markov's inequality, since it implies that the algorithm in Theorem 1 will still succeed with constant probability even when we stop after a time $\Theta(n/\sqrt{\gamma_f})$ if it has not succeeded so far. □

4 Classical complexity of random instances of BHSP

In this section we show that a uniformly chosen instance of BHSP is exponentially-hard classically with high probability.

Lemma 2. *A classical algorithm solving a uniformly random instance of BHSP with probability at least $1/2$ makes $\Omega(2^{n/2})$ oracle queries.*

Proof. Consider a classical algorithm $\mathcal{A}_{\text{cla}}$ that makes t_{cla} queries to the oracles A_f and A_g and with probability at least $1/2$ returns the unique s satisfying $g(x) \equiv f(x+s)$ (cf. Definition 1). For notational convenience we assume that $\mathcal{A}_{\text{cla}}$ only makes duplicated queries $(f(x), g(x))$. This can at most double the total number of oracle calls.

Consider the uniform distribution of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and $s \in \mathbb{Z}_2^n$, and let an input instance of BHSP be chosen accordingly. Let $(X_1, \ldots, X_{t_{\text{cla}}})$ be random variables representing the queries made by $\mathcal{A}_{\text{cla}}$. Then by the correctness assumption, the values $f(X_1), g(X_1), \ldots, f(X_{t_{\text{cla}}}), g(X_{t_{\text{cla}}})$ can be used to predict s with probability at least $1/2$.

First we observe that if, after k queries, it holds that $X_i - X_j \neq s$ for every $i, j \in [k]$, then even conditionally on the values of $f(X_1), g(X_1), \ldots, f(X_k), g(X_k)$ every $s \notin \{X_i - X_j | i, j \in [k]\}$ has exactly the same probability to occur. More precisely, if $S_k = \{X_i - X_j | i, j \in [k]\}$ and E_k is the event that $s \in S_k$, we have

$$\mathbf{Pr}\left[s = s_0 | \neg E_k\right] = \frac{1}{2^n - |S_k|} \leq \frac{1}{2^n - k^2} \tag{3}$$

for any $s_0 \notin S_k$ and $0 \leq k \leq t_{\text{cla}}$. In other words, modulo "$s \notin S_k$" the actual values of f and g at points $\{X_i | i \in [k]\}$ provide no additional information about s, and the best the algorithm can do in that case is a random guess, which succeeds with probability at most $1/(2^n - k^2)$.

Now let us analyze the probability that $S_{t_{\mathrm{cla}}} = \{X_i - X_j | i, j \in [t_{\mathrm{cla}}]\}$ contains s, that is, $\mathbf{Pr}\left[E_{t_{\mathrm{cla}}}\right]$. Since $|S_{k+1}| - |S_k| \leq k$, we have by the union bound

$$\mathbf{Pr}\left[E_{k+1} | \neg E_k\right] \leq \sum_{s_0 \in S_{k+1}} \mathbf{Pr}\left[s = s_0 | \neg E_k\right] \leq \frac{k}{2^n - k^2}.$$

Consequently,

$$\mathbf{Pr}\left[E_{t_{\mathrm{cla}}}\right] \leq \frac{\sum_{k=0}^{t_{\mathrm{cla}}-1} k}{2^n - t_{\mathrm{cla}}^2} \leq \frac{t_{\mathrm{cla}}^2}{2^n - t_{\mathrm{cla}}^2}.$$

Finally, we can bound the probability that the algorithm succeeds after t_{cla} oracle queries as

$$\begin{aligned}\mathbf{Pr}\left[\mathcal{A}_{\mathrm{cla}} \text{ succeeds}\right] &= \mathbf{Pr}\left[\mathcal{A}_{\mathrm{cla}} \text{ succeeds} | E_{t_{\mathrm{cla}}}\right] \cdot \mathbf{Pr}\left[E_{t_{\mathrm{cla}}}\right] \\ &\quad + \mathbf{Pr}\left[\mathcal{A}_{\mathrm{cla}} \text{ succeeds} | \neg E_{t_{\mathrm{cla}}}\right] \cdot \mathbf{Pr}\left[\neg E_{t_{\mathrm{cla}}}\right] \\ &\leq \mathbf{Pr}\left[E_{t_{\mathrm{cla}}}\right] + \mathbf{Pr}\left[\mathcal{A}_{\mathrm{cla}} \text{ succeeds} | \neg E_{t_{\mathrm{cla}}}\right] \leq \frac{t_{\mathrm{cla}}^2 + 1}{2^n - t_{\mathrm{cla}}^2},\end{aligned}$$

which is larger than $1/2$ only if $t_{\mathrm{cla}} \in \Omega\left(2^{n/2}\right)$, as required. □

We are now ready to state our main theorem which is an exponential quantum-classical separation for an average case problem.

Theorem 2 (Average case exponential separation). *Let $(\mathcal{O}_f, \mathcal{O}_g)$ be an instance of a Boolean hidden shift problem (BHSP) where $g(x) = f(x+v)$ and f and v are chosen uniformly at random. Then there exists a quantum algorithm which finds v with bounded error using $O(n)$ queries and in $O(\mathrm{poly}(n))$ time whereas any classical algorithm needs $\Omega(2^{n/2})$ queries to achieve the same task.*

Proof. For a fixed v and randomly chosen f, consider the 2^{n-1} mutually independent events "$f(x) = f(x+v)$". By definition of $\gamma_{f,v}$ and the Chernoff bound, the probability that $\gamma_{f,v} < 1/3$ is at most $e^{-\Omega(2^n)}$. Since this is double-exponentially small in n we obtain from an application of the union bound to the 2^n possible values of v that if $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is chosen uniformly at random then $\mathbf{Pr}_f[\gamma_f < 1/3] \in e^{-\Omega(2^n)}$. We now apply Corollary 1 for constant γ_f to obtain a quantum algorithm that uses at most $O(n)$ queries and outputs the correct hidden shift v with constant probability of success (i.e., ε is chosen to be constant). Combining this with the exponential lower bound from Lemma 2 implies that there is an exponential gap between the classical and quantum complexity of the BHSP defined over a random Boolean function. □

5 Discussion and open problems

We presented a quantum algorithm for the Boolean hidden shift problem that is based on sampling from the space of vectors that are orthogonal to the hidden

shift. It should be noted that our algorithm reduces to one of the two algorithms given in [Röt10] in case the function is a bent function. We related the running time and the query complexity of the algorithm to the minimum influence of the function and showed that for random functions these complexities are polynomial. This leads to an average case exponential separation between the classical and quantum time complexity for Boolean functions. An interesting question is whether these methods can be generalized and adapted for the case of non-Boolean functions also. Furthermore, we conjecture that the complexity of our quantum algorithm is optimal up to polynomial factors for any function.

Acknowledgments

The authors acknowledge support by ARO/NSA under grant W911NF-09-1-0569. We wish to thank Andrew Childs, Sean Hallgren, Guosen Yue and Ronald de Wolf for fruitful discussions.

References

[BHMT02] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In J. S. J. Lomonaco and H. E. Brandt, editors, *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *Contemporary Mathematics Series*, pages 53–74. American Mathematical Society, 2002. arXiv:quant-ph/0005055.

[BV97] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. (Earlier version in Proc. STOC'93, pp. 11–20). doi:10.1137/S0097539796300921.

[CM04] D. Curtis and D. A. Meyer. Towards quantum template matching. volume 5161, pages 134–141. SPIE, 2004. doi:10.1117/12.506669.

[CSV07] A. Childs, L. J. Schulman, and U. Vazirani. Quantum algorithms for hidden nonlinear structures. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 395–404, 2007. arXiv:0705.2784, doi:10.1109/FOCS.2007.18.

[CvD07] A. Childs and W. van Dam. Quantum algorithm for a generalized hidden shift problem. In *Proceedings of the 18th Symposium on Discrete Algorithms (SODA'07)*, pages 1225–1232, 2007. arXiv:quant-ph/0507190.

[CW07] A. Childs and P. Wocjan. On the quantum hardness of solving isomorphism problems as nonabelian hidden shift problems. *Quantum Information and Computation*, 7(5–6):504–521, 2007. arXiv:quant-ph/0510185.

[DMR10] H. Dinh, C. Moore, and A. Russell. The McEliece cryptosystem resists quantum Fourier sampling attacks. 2010. arXiv:1008.2390.

[FIM+03] K. Friedl, G. Ivanyos, F. Magniez, M. Santha, and P. Sen. Hidden translation and orbit coset in quantum computing. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC'03)*, pages 1–9, 2003.

[GOS+09] P. Gopalan, R. O'Donnell, R. Servedio, A. Shpilka, and K. Wimmer. Testing Fourier dimensionality and sparsity. In *36th International Colloquium on Automata, Languages and Programming (ICALP'09)*, volume 5555, pages 500–512, 2009. Long version: Carnegie Mellon Uni-

versity Technical Report (2009), *Computer Science Department*, Paper 1162. Available from: http://repository.cmu.edu/compsci/1162, doi:10.1007/978-3-642-02927-1_42.

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC'96)*, pages 212–219, New York, 1996. ACM. arXiv:quant-ph/9605043, doi:10.1145/237814.237866.

[Kup05] G. Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005. arXiv:quant-ph/0302112, doi:10.1137/S0097539703436345.

[Liu09] Y.-K. Liu. Quantum algorithms using the curvelet transform. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09)*, pages 391–400, New York, 2009. ACM. arXiv:0810.4968, doi:10.1145/1536414.1536469.

[MRRS07] C. Moore, D. Rockmore, A. Russell, and L. Schulman. The power of strong Fourier sampling: quantum algorithms for affine groups and hidden shifts. *SIAM Journal on Computing*, 37(3):938–958, 2007. arXiv:quant-ph/0503095, doi:10.1137/S0097539705447177.

[Reg04] O. Regev. Quantum computation and lattice problems. *SIAM Journal on Computing*, 33(2):738–760, 2004. arXiv:cs/0304005, doi:10.1137/S0097539703440678.

[Röt10] M. Rötteler. Quantum algorithms for highly non-linear Boolean functions. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, pages 448–457, 2010. arXiv:0811.3208.

[vD08] W. van Dam. Quantum algorithms for weighing matrices and quadratic residues. *Algorithmica*, 34:413–428, 2008. arXiv:quant-ph/0008059, doi:10.1007/s00453-002-0975-4.

[vDHI06] W. van Dam, S. Hallgren, and L. Ip. Quantum algorithms for some hidden shift problems. *SIAM Journal on Computing*, 36:763–778, 2006. (Earlier version in Proc. SODA'03, pp.489–498). arXiv:quant-ph/0211140, doi:10.1137/S009753970343141X.

A Proof of Lemma 1

Lemma 1 $\gamma_{f,v} = \sum_{u:\langle v,u\rangle=1} \left|\widehat{F}(u)\right|^2$.

Proof. Let us consider the following function $\widetilde{F_v}(x) \stackrel{\text{def}}{=} F(x) - F(x+v)$. Its Fourier transform reads

$$\widehat{\widetilde{F_v}}(u) = \mathop{\mathbf{E}}_x \left[F(x)\cdot\chi_u(x) - F(x+v)\cdot\chi_u(x)\right] = (1-\chi_u(v))\cdot\widehat{F}(u).$$

Therefore, we have

$$\begin{aligned}\sum_{u:\langle v,u\rangle=1}\left|\widehat{F}(u)\right|^2 &= \frac{1}{4}\sum_{u\in\mathbb{Z}_2^n}\left|(1-\chi_u(v))\cdot\widehat{F}(u)\right|^2 = \frac{1}{4}\sum_{u\in\mathbb{Z}_2^n}\left|\widehat{\widetilde{F_v}}(u)\right|^2 \\ &= \frac{1}{4}\mathop{\mathbf{E}}_x\left[\left|\widetilde{F_v}(x)\right|^2\right] = \mathop{\mathbf{Pr}}_x\left[F(x)\neq F(x+v)\right] = \gamma_{f,v},\end{aligned}$$

where in the second line we have used Parseval's identity. □